**Part 22**

1. ASCII(Character\_Expression) - Returns the ASCII code of the given character expression.

2. CHAR(Integer\_Expression) - Converts an int ASCII code to a character. The Integer\_Expression, should be between 0 and 255.

3. LTRIM(Character\_Expression) - Removes blanks on the left handside of the given character expression.

4. RTRIM(Character\_Expression) - Removes blanks on the right hand side of the given character expression.

5. LOWER(Character\_Expression) - Converts all the characters in the given Character\_Expression, to lowercase letters.

6. UPPER(Character\_Expression) - Converts all the characters in the given Character\_Expression, to uppercase letters.

7. REVERSE('Any\_String\_Expression') - Reverses all the characters in the given string expression.

8. LEN(String\_Expression) - Returns the count of total characters, in the given string expression, excluding the blanks at the end of the expression.

**Part 23**

LEFT(Character\_Expression, Integer\_Expression) - Returns the specified number of characters from the left hand side of the given character expression.

Example: Select LEFT('ABCDE', 3)

Output: ABC

RIGHT(Character\_Expression, Integer\_Expression) - Returns the specified number of characters from the right hand side of the given character expression.

Example: Select RIGHT('ABCDE', 3)

Output: CDE

CHARINDEX('Expression\_To\_Find', 'Expression\_To\_Search', 'Start\_Location') - Returns the starting position of the specified expression in a character string. Start\_Location parameter is optional.

Example: In this example, we get the starting position of '@' character in the email string 'sara@aaa.com'.

Select CHARINDEX('@','sara@aaa.com',1)

Output: 5

SUBSTRING('Expression', 'Start', 'Length') - As the name, suggests, this function returns substring (part of the string), from the given expression. You specify the starting location using the 'start' parameter and the number of characters in the substring using 'Length' parameter. All the 3 parameters are mandatory.

Example: Display just the domain part of the given email 'John@bbb.com'.

Select SUBSTRING('John@bbb.com',6, 7)

Output: bbb.com

In the above example, we have hardcoded the starting position and the length parameters. Instead of hardcoding we can dynamically retrieve them using CHARINDEX() and LEN() string functions as shown below.

Example:

Select SUBSTRING('John@bbb.com',(CHARINDEX('@', 'John@bbb.com') + 1), (LEN('John@bbb.com') - CHARINDEX('@','John@bbb.com')))

Output: bbb.com

**Part 24**  
**REPLICATE(String\_To\_Be\_Replicated, Number\_Of\_Times\_To\_Replicate)** - Repeats the given string, for the specified number of times.  
  
**Example**: SELECT REPLICATE('Pragim', 3)  
**Output**: Pragim Pragim Pragim

**SPACE(Number\_Of\_Spaces)** - Returns number of spaces, specified by the Number\_Of\_Spaces argument.  
  
**Example**: The SPACE(5) function, inserts 5 spaces between FirstName and LastName  
Select FirstName + SPACE(5) + LastName as FullName  
From tblEmployee

**PATINDEX('%Pattern%', Expression)**  
Returns the starting position of the first occurrence of a pattern in a specified expression. It takes two arguments, the pattern to be searched and the expression. PATINDEX() is simial to CHARINDEX(). With CHARINDEX() we cannot use wildcards, where as PATINDEX() provides this capability. If the specified pattern is not found, PATINDEX() returns ZERO.  
  
**Example:**   
Select Email, PATINDEX('%@aaa.com', Email) as FirstOccurence   
from tblEmployee  
Where PATINDEX('%@aaa.com', Email) > 0

**REPLACE(String\_Expression, Pattern , Replacement\_Value)**  
Replaces all occurrences of a specified string value with another string value.  
  
**Example**: All .COM strings are replaced with .NET  
Select Email, REPLACE(Email, '.com', '.net') as ConvertedEmail  
from  tblEmployee

**STUFF(Original\_Expression, Start, Length, Replacement\_expression)**  
STUFF() function inserts Replacement\_expression, at the start position specified, along with removing the charactes specified using Length parameter.  
  
**Example**:  
Select FirstName, LastName,Email, STUFF(Email, 2, 3, '\*\*\*\*\*') as StuffedEmail  
From tblEmployee